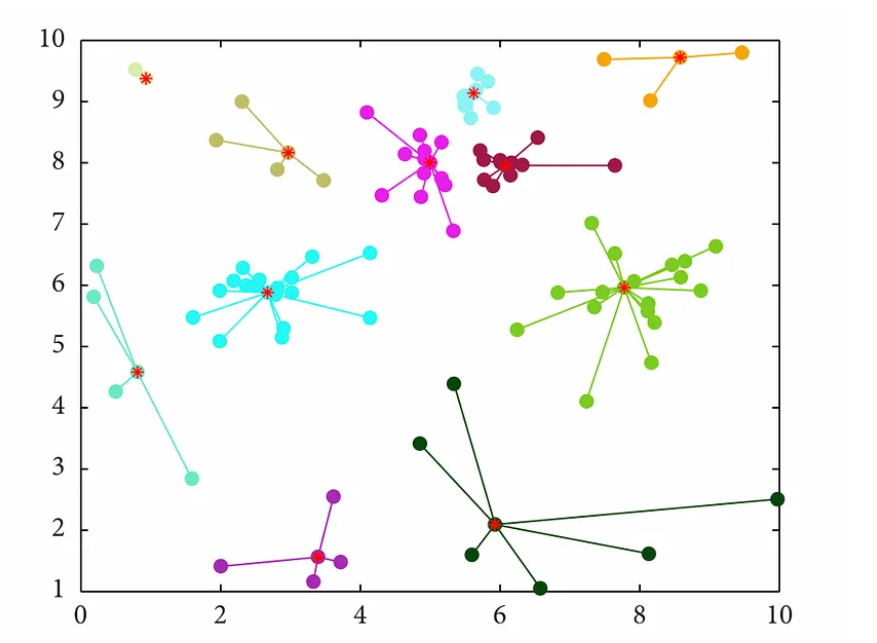
**Affinity Propagation:**

Affinity Propagation is a clustering algorithm that identifies a set of exemplars among the data points and forms clusters around these exemplars. Unlike other clustering methods that require the user to specify the number of clusters in advance, it automatically determines the number of clusters based on the data.

Exemplars are specific data points that serve as representatives or prototypes for each cluster. Essentially, an exemplar is a data point within a cluster that best summarizes or characterizes the other data points in that cluster.

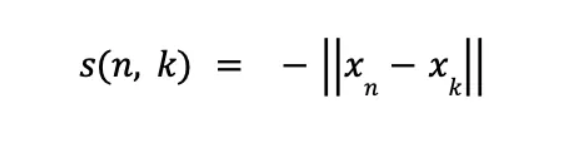
****

K-means, although sometimes referred to as an exemplar-based clustering method, fundamentally differs from Affinity Propagation in this regard. In K-means, the centroids that represent clusters are computed based on the mean of the data points within each cluster, and they may not coincide with actual data points. This distinction highlights one of the key differences between K-means and Affinity Propagation: while Affinity Propagation chooses exemplars from the existing data points, K-means calculates centroids that may not correspond to any specific data point, resulting in a different approach to characterizing and forming clusters.

k(n): exemplar of n. If k is the exemplar of n then k(n) = k.

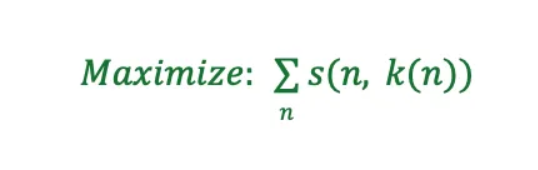
**Similarity**

The similarity *s(n, k)* between two data points is typically defined as the negative Euclidean distance between them.



The similarity measure captures how well-suited data point k is to be an exemplar for data point n. A higher similarity value indicates that the two points are closer or more similar to each other.

**The objective of the Affinity Propagation clustering algorithm is to maximize the summation over every data point n of the similarity between n and its exemplar k(n).**

****

For a data point to be considered the exemplar for other data points, it must first be an exemplar for itself.

The similarity measure plays a critical role in the “responsibility” and “availability” message updates, which are core to the iterative process of Affinity Propagation. These messages are updated based on the similarities and current estimates of responsibilities and availabilities.

## Responsibility

Responsibility is one of the two types of messages exchanged between data points to decide on the most suitable exemplars for clusters.

The responsibility ρ(n, k) sent from data point n to candidate exemplar k reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point n, taking into account other potential exemplars for n.

s(n, k) term represents the similarity between data points n and k. max term finds the maximum value among all potential exemplars l (excluding the current candidate k). For each potential exemplar l, it sums the availability α and similarity s.

By subtracting the maximum value from the similarity, we are essentially measuring how much better (or worse) k is as an exemplar for n compared to the best-competing candidate.

The responsibility message plays a pivotal role in the iterative process of the algorithm. It provides feedback from data points to candidate exemplars about their suitability. If a particular point k consistently receives high-responsibility messages from many data points, it’s more likely to be selected as an exemplar.

A high positive responsibility value from point n to point k indicates that k is a particularly good fit to be the exemplar for n, given the other potential exemplars.

## Availability

It is the second type of message that gets exchanged between data points. Availability provides feedback to a data point regarding its suitability to be an exemplar based on the accumulated evidence from other data points.

The availability message provides feedback to individual data points about their suitability to be exemplars, considering the collective opinion of other points.

## Complete Process

* Each data point is treated as a potential exemplar.
* The responsibility and availability matrices are initialized to 0.
* The algorithm proceeds iteratively, with data points exchanging responsibility and availability messages until either convergence is reached or a set number of iterations is completed.
* In each iteration: The responsibility matrix is updated based on the current similarity and availability values. Subsequently, the availability matrix is updated based on the newly computed responsibility values.
* After updating the matrices in each iteration, the sum of the responsibility and availability for each data point is computed.
* For each data point n, if the value ρ(n, n) + α(n, n) is positive, then n is selected as an exemplar for itself. This means that the data point believes it should be an exemplar and has received enough corroborative feedback from other points to support this belief.
* Once the exemplars are determined, each non-exemplar data point n is assigned to an exemplar k that maximizes the sum ρ(n, k) + α(n, k). This means each non-exemplar point is assigned to the exemplar to which it feels the strongest affinity.
* The iterative process stops when either the exemplar assignments remain unchanged over a specified number of iterations or a predefined maximum number of iterations is reached.
* At the end of the process, the algorithm provides a set of exemplars and the assignment of all other data points to these exemplars, effectively partitioning the data into clusters.

Parameters:

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html

* damping is a factor between 0.5 and 1 that down-weights the influence of incoming messages when updating messages. This helps to avoid numerical oscillations when updating the messages.
* max\_iter is the maximum number of iterations.
* convergence\_iter is the number of iterations with no change in the number of estimated clusters that stops the convergence. If the cluster assignments remain unchanged for this many consecutive iterations, the algorithm terminates.
* copy copies the data if it is set to True.
* preference is the “preference” value for each data point. If set to None, the median of the input similarities is used. It influences the number of exemplars the algorithm will find. **A lower value results in more clusters.**
* **Preference array-like of shape (n\_samples,) or float, default=None:** Preferences for each point - points with larger values of preferences are more likely to be chosen as exemplars. The number of exemplars, ie of clusters, is influenced by the input preferences value. If the preferences are not passed as arguments, they will be set to the median of the input similarities.
* affinity specifies which affinity (similarity measure) to use. euclidean is a negative squared Euclidean distance between data points. precomputed is the input data provided as a precomputed similarity matrix.

Strengths: The user doesn't need to specify the number of clusters (but does need to specify 'sample preference' and 'damping' hyperparameters).

Weaknesses: The main disadvantage of Affinity Propagation is that it's quite slow and memory-heavy, making it difficult to scale to larger datasets